## CLAIMS

1. Device for sharing and controlling access to peripherals for a computer system comprising a central processor (CPU) and at least one input/output peripheral having a physical control interface accessible to the central processor, characterised in that said device has:

- means for the faithful reproduction, in the form of a virtual interface, of the physical interface of at least one peripheral,
- means of interception, by said virtual interface, of all the requests and data exchanged between the central processor and the peripheral, controlled by a pre-determined application executed in the system,
- means of possible modification of said requests and data intercepted according to at least one pre-determined criterion.

2. Device according to Claim 1, characterised in that the means of reproducing this physical interface (9A) in virtual form comprise:

- a memory space (131, 141) reserved for the image of the physical interface, peculiar to each application executed by the computer system,
- a means for linking the addresses of these memory spaces (131, 141) to the physical interface address (9A).

3. Device according to either one of Claims 1 to 2, characterised in that the interception means comprise:

- on the one hand an interface (21) with the bus (2') connected to the central processing unit (3), and an interface (23) with the bus (2) connected to the peripherals (6, 9, 10),
- and on the other hand an address decoding means (24).

4. Device according to any one of Claims 1 to 3, characterised in that the modification means comprise a means of filtering the requests intercepted by the interception means, according to at least one criterion stored in a memory means (25).

5. Device according to any one of Claims 1 to 4, characterised in that it is composed of:

- a module (16) inserted between the central processing unit (3) and the peripherals bus (2), and
- a software element previously stored in a memory means of the central processing unit (3).

6. Device according to Claim 5, characterised in that the module (16) has:

- an input/output bus interface (21) connected by the processor bus (2') to the pair formed by the central processing unit (3) and the memory (4) by means of the bridge (19),
- a programmable logic unit (22),
- an input/output bus interface (23) connected to the address and data bus (2).

7. Device according to any one of Claims 5 to 6, characterised in that the programmable logic unit (22) has an address decoder (24), a local memory (25) and a programmable filter (26).

8. Device according to Claim 7, characterised in that it has means of insertion in the interface of the primary communication bus (2') connected to the random access memory (4).

9. Device according to any one of Claims 7 to 8, characterised in that the address decoder (24) has means of selecting at least one filtering pattern for the data included in a request, according to the address decoded in the request.

10. Device according to any one of Claims 7 to 9, characterised in that the programmable filter (26) has means adapted to apply, to the data included in the requests, predetermined filtering patterns constituting criteria for checking the integrity of the system.

11. Device according to any one of Claims 5 to 10, characterised in that it has means adapted so that, when the system is initialised, for each application (13, 14) liable to request access in read or write mode to a particular peripheral (9A), present in the operating system (12) downstream of the module

(16), the operating system (12) installs, in the virtual memory space (130) of the application (13, 14), an access (133, 143) to the physical memory (4) in a particular area (131, 141) referred to as the virtual io-pages area of the module (16).

12. Device according to Claim 11, characterised in that the size of the virtual io-pages area (131, 141) is equivalent to the memory space occupied by the physical interface (9A) of the peripheral (9) in question.

13. Device according to either one of Claims 11 to 12, characterised in that it has means adapted so that the operating system (12) initialises, for each application (13, 14), a vector field (160, 161) specific to each application in the local memory (25) of the module (16), specifying the addresses for translation of the virtual io-pages (131, 141) into physical io-pages which are integrated into the physical interface of the peripheral (9A).

14. Device according to any one of Claims 11 to 13, characterised in that it has means adapted so that the operating system (12) initialises, for each application, an area (132, 142) of the local memory (25) of the module (16), equivalent to the decoding area (131, 141), with the filtering patterns to be applied to each access of the application (13, 14).

15. Device according to Claim 14, characterised in that it has means adapted so that

- when the computer system is started up, the operating system (12) initialises the local memory (25) of the module (16), sending to it
  - the filtering patterns to be applied to the different virtual io-pages addresses in read or write mode for the shared peripherals,
  - the translation between the addresses of the virtual io-pages (141, 131) and those of the corresponding physical io-pages in the physical interface (9A)
- the module (16) waits until it receives a request from an application (13, 14) in read or write mode to the shared peripherals at the virtual io-pages addresses (131, 141),

- in the case of a write command coming from the central processing unit (3), the data item is modified and then applied to the address and data bus (2) on the peripherals side,

- in the case of a command in read mode, the request is transmitted to the peripheral, and then the module (16) awaits a response from said peripheral, the data item to be modified then being the one coming from the bus (2) on the peripherals side, this data item is then modified, and then the data item once modified is applied to the bus of the processor (2') at the central processing unit (3).

16. Method of sharing and controlling access to peripherals for a computer system comprising a central processor (CPU) and at least one input/output peripheral having a physical control interface accessible to the central processor, characterised in that it includes:

- a step of reproducing, in the form of a virtual interface, the physical interface of at least one peripheral,

- a step of interception by said virtual interface of all the requests and data exchanged between the central processor and the peripheral, controlled by a predetermined application executed in the system,

- a step of possible modification of said requests and data intercepted according to at least one predetermined criterion.

17. Method according to Claim 16, characterised in that the step of reproducing this physical interface (9A) in virtual form comprises the creation of:

- a memory space (131, 141) reserved for the image of the physical interface (9A), peculiar to each application executed by the computer system,

- a mechanism for linking the physical addresses (131, 141) of these memory spaces to the address of the interface (9A)

- a field (132) specifying the filtering functions to be applied to the memory area (131).

18. Method according to either one of Claims 16 to 17, characterised in that it comprises a step of selecting at least one filtering pattern

for the data included in a request, according to the decoded address in the request.

19. Method according to Claim 18, characterised in that it includes a step of applying, to the data included in the request, predetermined filtering patterns constituting criteria for checking the integrity of the system.

20. Method according to any one of Claims 16 to 19, characterised in that it includes a step, during the initialisation of the system, for each application (13, 14) liable to request access in read mode or write mode to a particular peripheral (9A), present in the operating system (12) downstream of the module (16), for installation by the operating system (12) in the virtual memory space (130) of the application (13, 14) of an access (133, 143) to the physical memory (4) in a particular area (131, 141) referred to as the decoding area of the module (16).

21. Method according to Claim 20, characterised in that the size of the decoding area (131, 141) is equivalent to the memory space occupied by the physical interface (9A) of the peripheral (9) in question.

22. Method according to any one of Claims 16 to 21, characterised in that it includes a step of initialisation, by the operating system (12), for each application (13, 14), of a vector field (160, 161) specific to each application in the local memory (25) of the module (16), specifying the addresses for translation of the virtual io-pages (131, 141) into physical io-pages which are integrated into the physical interface of the peripheral (9A).

23. Method according to any one of Claims 20 to 22, characterised in that it includes a step of initialisation by the operating system (12) for each application of an area (132, 142) of the local memory (25) of the module (16), equivalent to the decoding area (131, 141), with the filtering patterns to be applied to each access of the application (13, 14).

24. Method according to Claim 23, characterised in that it includes steps such that:

- in a first step (E1), when the computer system is started up, the operating system (12) initialises the local memory (25) of the module (16), sending to it

- the filtering patterns to be applied to the different virtual io-pages addresses in read or write mode for the shared peripherals,
- the translation between the addresses of the virtual io-pages (141, 131) and those of the corresponding physical io-pages in the physical interface (9A)

- in a step (E2), the module (16) waits until it receives a request from an application (13, 14) in read or write mode to the shared peripherals at the virtual io-pages addresses (131, 141),

- in the case of a write command coming from the central processing unit (3), the data item is modified in a step (E3) and then applied to the address and data bus (2) on the peripherals side in a step (E4),

- in the case of a read command, the request is transmitted to the peripheral in a step (E5), and then the module (16) awaits a response from said peripheral in a step (E6), the data item to be modified then being the one coming from the bus (2) on the peripherals side, this data item is then modified in a step (E7), and then the data item once modified is applied to the bus of the processor (2') at the central processing unit (3) in a step (E8).

25. An information storage means which is removable, partially or totally, and which can be read by a computer or a microprocessor storing portions of code of a computer program, characterized in that it makes it possible to implement the method according to any one of the preceding claims 16 to 24.

26. A computer program product which can be loaded into a programmable apparatus, containing portions of code for implementing the steps of the method according to any one of the preceding claims 16 to 24, when the program is executed on a programmable apparatus.